



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	2MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	5
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mchlc908qt4pe

Table of Contents

Chapter 1 General Description

1.1	Introduction	15
1.2	Features	15
1.3	MCU Block Diagram	17
1.4	Pin Assignments	17
1.5	Pin Functions	20
1.6	Pin Function Priority	21

Chapter 2 Memory

2.1	Introduction	23
2.2	Unimplemented Memory Locations	23
2.3	Reserved Memory Locations	23
2.4	Input/Output (I/O) Section	25
2.5	Random-Access Memory (RAM)	31
2.6	FLASH Memory (FLASH)	32
2.6.1	FLASH Control Register	32
2.6.2	FLASH Page Erase Operation	33
2.6.3	FLASH Mass Erase Operation	34
2.6.4	FLASH Program Operation	34
2.6.5	FLASH Protection	35
2.6.6	FLASH Block Protect Register	37
2.6.7	Wait Mode	38
2.6.8	Stop Mode	38

Chapter 3 Analog-to-Digital Converter (ADC)

3.1	Introduction	39
3.2	Features	39
3.3	Functional Description	39
3.3.1	ADC Port I/O Pins	39
3.3.2	Voltage Conversion	41
3.3.3	Conversion Time	42
3.3.4	Continuous Conversion	42
3.3.5	Accuracy and Precision	42
3.4	Interrupts	42
3.5	Low-Power Modes	42
3.5.1	Wait Mode	42
3.5.2	Stop Mode	42

Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0005	Data Direction Register B (DDRB) See page 101.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0006 ↓ \$000A	Unimplemented									
\$000B	Port A Input Pullup Enable Register (PTAPUE) See page 99.	Read:	OSC2EN	0	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000C	Port B Input Pullup Enable Register (PTBPUE) See page 102.	Read:	PTBPUE7	PTBPUE6	PTBPUE5	PTBPUE4	PTBPUE3	PTBPUE2	PTBPUE1	PTBPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000D ↓ \$0019	Unimplemented									
\$001A	Keyboard Status and Control Register (KBSCR) See page 83.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER) See page 84.	Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$001C	Unimplemented									
\$001D	IRQ Status and Control Register (INTSCR) See page 77.	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0
\$001E	Configuration Register 2 (CONFIG2) ⁽¹⁾ See page 53.	Read:	IRQPUD	IRQEN	R	OSCOPT1	OSCOPT0	R	R	RSTEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0 ⁽²⁾

1. One-time writable register after each reset.

2. RSTEN reset to 0 by a power-on reset (POR) only.


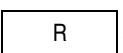
 = Unimplemented  = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 6)

MASS — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation.

1 = Mass Erase operation selected

0 = Mass Erase operation unselected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Erase operation selected

0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

1 = Program operation selected

0 = Program operation unselected

2.6.2 FLASH Page Erase Operation

Use the following procedure to erase a page of FLASH memory. A page consists of 64 consecutive bytes starting from addresses \$XX00, \$XX40, \$XX80, or \$XXC0. The 48-byte user interrupt vectors area also forms a page. Any FLASH memory page can be erased alone.

1. Set the ERASE bit and clear the MASS bit in the FLASH control register.
2. Read the FLASH block protect register.
3. Write any data to any FLASH location within the address range of the block to be erased.
4. Wait for a time, t_{NVS} (minimum 10 μ s).
5. Set the HVEN bit.
6. Wait for a time, t_{Erase} (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for a time, t_{NVH} (minimum 5 μ s).
9. Clear the HVEN bit.
10. After time, t_{RCV} (typical 1 μ s), the memory can be accessed in read mode again.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

CAUTION

A page erase of the vector page will erase the internal oscillator trim value at \$FFC0.

In applications that require more than 1000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a shorter cycle time.

2.6.6 FLASH Block Protect Register

The FLASH block protect register is implemented as a byte within the FLASH memory, and therefore can only be written during a programming sequence of the FLASH memory. The value in this register determines the starting address of the protected range within the FLASH memory.

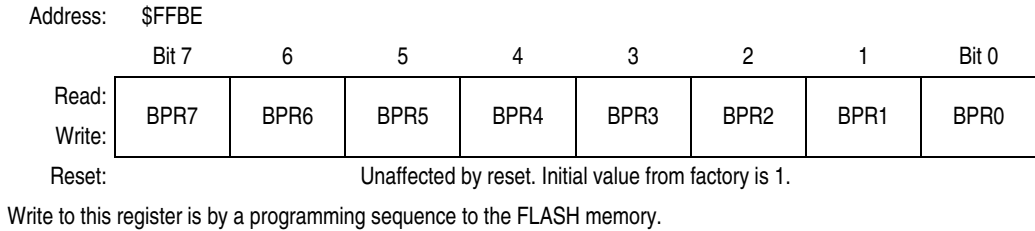


Figure 2-5. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Protection Register Bits [7:0]

These eight bits in FLBPR represent bits [13:6] of a 16-bit memory address. Bits [15:14] are 1s and bits [5:0] are 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at \$FFFF. With this mechanism, the protect start address can be XX00, XX40, XX80, or XXC0 within the FLASH memory. See Figure 2-6 and Table 2-2.

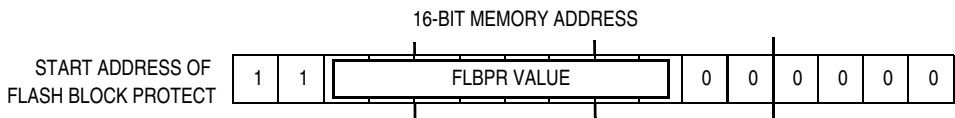


Figure 2-6. FLASH Block Protect Start Address

Table 2-2. Examples of Protect Start Address

BPR[7:0]	Start of Address of Protect Range
\$00–\$B8	The entire FLASH memory is protected.
\$B9 (1011 1001)	\$EE40 (1110 1110 0100 0000)
\$BA (1011 1010)	\$EE80 (1110 1110 1000 0000)
\$BB (1011 1011)	\$EEC0 (1110 1110 1100 0000)
\$BC (1011 1100)	\$EF00 (1110 1111 0000 0000)
and so on...	
\$DE (1101 1110)	\$F780 (1111 0111 1000 0000)
\$DF (1101 1111)	\$F7C0 (1111 0111 1100 0000)
\$FE (1111 1110)	\$FF80 (1111 1111 1000 0000) FLBPR, OSCTRIM, and vectors are protected
\$FF	The entire FLASH memory is not protected.

3.3.3 Conversion Time

Sixteen ADC internal clocks are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADSCR. If the ADC internal clock is selected to run at 1 MHz, then one conversion will take 16 μ s to complete. With a 1-MHz ADC internal clock the maximum sample rate is 62.5 kHz.

$$\text{Conversion Time} = \frac{16 \text{ ADC Clock Cycles}}{\text{ADC Clock Frequency}}$$

$$\text{Number of Bus Cycles} = \text{Conversion Time} \times \text{Bus Frequency}$$

3.3.4 Continuous Conversion

In the continuous conversion mode (ADCO = 1), the ADC continuously converts the selected channel filling the ADC data register (ADR) with new data after each conversion. Data from the previous conversion will be overwritten whether that data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit (ADSCR, \$003C) is set after each conversion and will stay set until the next read of the ADC data register.

When a conversion is in process and the ADSCR is written, the current conversion data should be discarded to prevent an incorrect reading.

3.3.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

3.4 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a central processor unit (CPU) interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

3.5 Low-Power Modes

The following subsections describe the ADC in low-power modes.

3.5.1 Wait Mode

The ADC continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the microcontroller unit (MCU) out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting the CH[4:0] bits in ADSCR to 1s before executing the WAIT instruction.

3.5.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode. Allow one conversion cycle to stabilize the analog circuitry before using ADC data after exiting stop mode.

4.4 Wait Mode

The AWU module remains inactive in wait mode.

4.5 Stop Mode

When the AWU module is enabled (AWUIE = 1 in the keyboard interrupt enable register) it is activated automatically upon entering stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode. The AWU counters start from '0' each time stop mode is entered.

4.6 Input/Output Registers

The AWU shares registers with the keyboard interrupt (KBI) module and the port A I/O module. The following I/O registers control and monitor operation of the AWU:

- Port A data register (PTA)
- Keyboard interrupt status and control register (KBSCR)
- Keyboard interrupt enable register (KBIER)

4.6.1 Port A I/O Register

The port A data register (PTA) contains a data latch for the state of the AWU interrupt request, in addition to the data latches for port A.

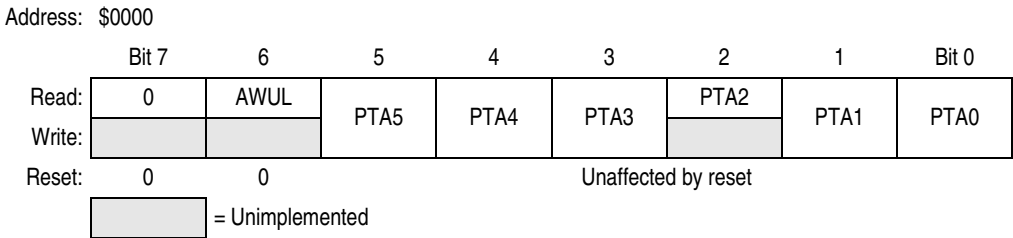


Figure 4-2. Port A Data Register (PTA)

AWUL — Auto Wakeup Latch

This is a read-only bit which has the value of the auto wakeup interrupt request latch. The wakeup request signal is generated internally. There is no PTA6 port or any of the associated bits such as PTA6 data direction or pullup bits.

- 1 = Auto wakeup interrupt request is pending
- 0 = Auto wakeup interrupt request is not pending

NOTE

PTA5–PTA0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [12.2.1 Port A Data Register](#).

Chapter 7

Central Processor Unit (CPU)

7.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

7.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

7.3 CPU Registers

[Figure 7-1](#) shows the five CPU registers. CPU registers are not part of the memory map.

Keyboard Interrupt Module (KBI)

9.7.2 Keyboard Interrupt Enable Register

The port A keyboard interrupt enable register (KBIER) enables or disables each port A pin or auto wakeup to operate as a keyboard interrupt input.

Address: \$001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 9-4. Keyboard Interrupt Enable Register (KBIER)

KBIE5–KBIE0 — Port A Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin on port A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

- 1 = KBIx pin enabled as keyboard interrupt pin
- 0 = KBIx pin not enabled as keyboard interrupt pin

NOTE

AWUIE bit is not used in conjunction with the keyboard interrupt feature. To see a description of this bit, see [Chapter 4 Auto Wakeup Module \(AWU\)](#).

11.6 Oscillator During Break Mode

The oscillator continues to drive BUSCLKX2 and BUSCLKX4 when the device enters the break state.

11.7 CONFIG2 Options

Two CONFIG2 register options affect the operation of the oscillator module: OSCOPT1 and OSCOPT0. All CONFIG2 register bits will have a default configuration. Refer to [Chapter 5 Configuration Register \(CONFIG\)](#) for more information on how the CONFIG2 register is used.

Table 11-2 shows how the OSCOPT bits are used to select the oscillator clock source.

Table 11-2. Oscillator Modes

OSCOPT1	OSCOPT0	Oscillator Modes
0	0	Internal Oscillator
0	1	External Oscillator
1	0	External RC
1	1	External Crystal

11.8 Input/Output (I/O) Registers

The oscillator module contains these two registers:

1. Oscillator status register (OSCSTAT)
2. Oscillator trim register (OSCTRIM)

11.8.1 Oscillator Status Register

The oscillator status register (OSCSTAT) contains the bits for switching from internal to external clock sources.

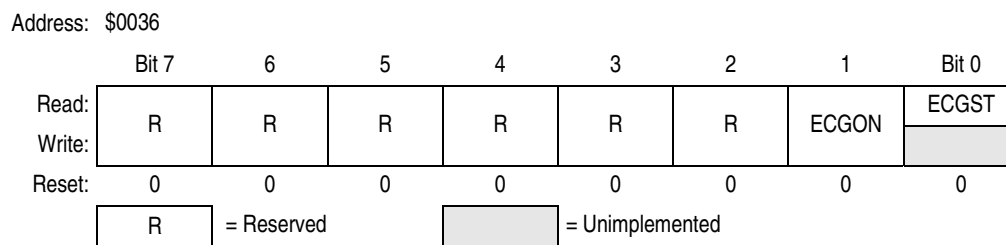


Figure 11-4. Oscillator Status Register (OSCSTAT)

ECGON — External Clock Generator On Bit

This read/write bit enables external clock generator, so that the switching process can be initiated. This bit is forced low during reset. This bit is ignored in monitor mode with the internal oscillator bypassed.

- 1 = External clock generator enabled
- 0 = External clock generator disabled

ECGST — External Clock Status Bit

This read-only bit indicates whether or not an external clock source is engaged to drive the system clock.

- 1 = An external clock source engaged
- 0 = An external clock source disengaged

Chapter 12

Input/Output Ports (PORTS)

12.1 Introduction

The MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 have five bidirectional input-output (I/O) pins and one input only pin. The MC68HLC908QY1, MC68HLC908QY2, and MC68HLC908QY4 have thirteen bidirectional pins and one input only pin. All I/O pins are programmable as inputs or outputs.

NOTE

Connect any unused I/O pins to an appropriate logic level, either V_{DD} or V_{SS} . Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.

8-pin devices have non-bonded pins. These pins should be configured either as outputs driving low or high, or as inputs with internal pullups enabled. Configuring these non-bonded pins in this manner will prevent any excess current consumption caused by floating inputs.

12.2 Port A

Port A is a 6-bit special function port that shares all six of its pins with the keyboard interrupt (KBI) module (see [Chapter 9 Keyboard Interrupt Module \(KBI\)](#)). Each port A pin also has a software configurable pullup device if the corresponding port pin is configured as an input port.

NOTE

PTA2 is input only.

When the \overline{IRQ} function is enabled in the configuration register 2 (CONFIG2), bit 2 of the port A data register (PTA) will always read a 0. In this case, the BIH and BIL instructions can be used to read the logic level on the PTA2 pin. When the \overline{IRQ} function is disabled, these instructions will behave as if the PTA2 pin is a logic 1. However, reading bit 2 of PTA will read the actual logic level on the pin.

13.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter (see [13.7.2 Stop Mode](#) for details.) The SIM counter is free-running after all reset states. See [13.4.2 Active Resets from Internal Sources](#) for counter control and internal reset recovery sequences.

13.6 Exception Control

Normal sequential program execution can be changed in three different ways:

1. Interrupts
 - a. Maskable hardware CPU interrupts
 - b. Non-maskable software interrupt instruction (SWI)
2. Reset
3. Break interrupts

13.6.1 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. [Figure 13-7](#) flow charts the handling of system interrupts.

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared).

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. [Figure 13-8](#) shows interrupt entry timing. [Figure 13-9](#) shows interrupt recovery timing.

13.6.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register), and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If more than one interrupt is pending at the end of an instruction execution, the highest priority interrupt is serviced first. [Figure 13-10](#) demonstrates what happens when two interrupts are pending. If an interrupt is pending upon exit from the original interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

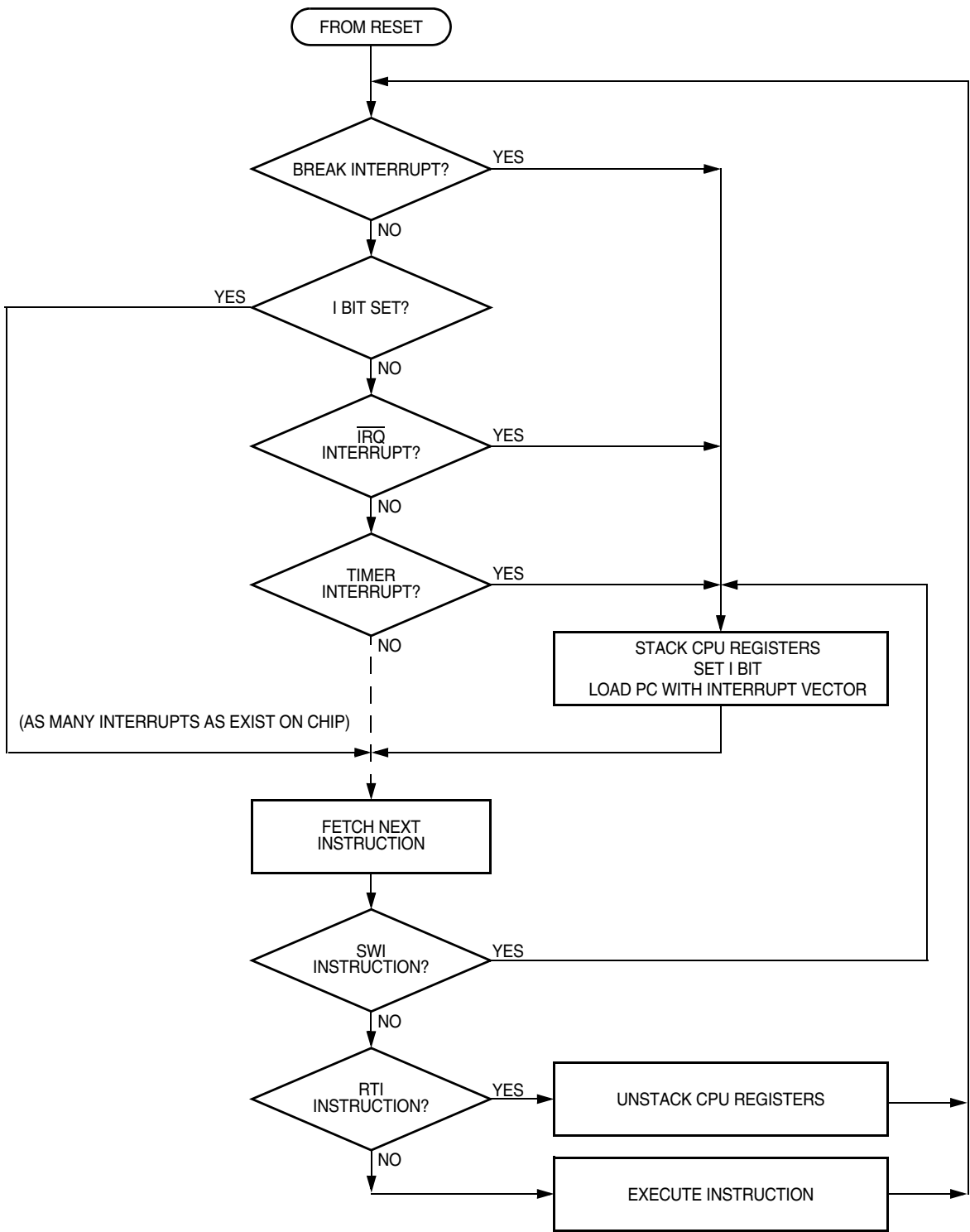


Figure 13-7. Interrupt Processing

14.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [14.4.4 Pulse Width Modulation \(PWM\)](#). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

14.4.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE

In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.

14.6 Wait Mode

The WAIT instruction puts the MCU in low power-consumption standby mode.

The TIM remains active after the execution of a WAIT instruction. In wait mode the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

14.7 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See [13.8.2 Break Flag Control Register](#).

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

14.8 Input/Output Signals

Port A shares three of its pins with the TIM. Two TIM channel I/O pins are PTA0/TCH0 and PTA1/TCH1 and an alternate clock source is PTA2/TCLK.

14.8.1 TIM Clock Pin (PTA2/TCLK)

PTA2/TCLK is an external clock input that can be the clock source for the TIM counter instead of the prescaled internal bus clock. Select the PTA2/TCLK input by writing 1s to the three prescaler select bits, PS[2–0]. (See [14.9.1 TIM Status and Control Register](#).) When the PTA2/TCLK pin is the TIM clock input, it is an input regardless of port pin initialization.

14.8.2 TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. PTA0/TCH0 can be configured as a buffered output compare or buffered PWM pin.

14.9 Input/Output Registers

The following I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)

Setting MS0B disables the channel 1 status and control register and reverts TCH1 to general-purpose I/O.

Reset clears the MSxB bit.

- 1 = Buffered output compare/PWM operation enabled
- 0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation.

See [Table 14-3](#).

- 1 = Unbuffered output compare/PWM operation
- 0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin (see [Table 14-3](#)).

Reset clears the MSxA bit.

- 1 = Initial output level low
- 0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM status and control register (TSC).

Table 14-3. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

- 1 = Status bits clearable during break
- 0 = Status bits not clearable during break

15.2.3 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power- consumption standby modes. If enabled, the break module will remain enabled in wait and stop modes. However, since the internal address bus does not increment in these modes, a break interrupt will never be triggered.

15.3 Monitor Module (MON)

This subsection describes the monitor module (MON) and the monitor mode entry methods. The monitor allows debugging and programming of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST} , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features include:

- Normal user-mode pin functionality on most pins
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Execution of code in random-access memory (RAM) or FLASH
- FLASH memory security feature⁽¹⁾
- FLASH memory programming interface
- Use of external 9.8304 MHz oscillator to generate internal frequency of 2.4576 MHz
- Simple internal oscillator mode of operation (no external clock or high voltage)
- Monitor mode entry without high voltage, V_{TST} , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Standard monitor mode entry if high voltage is applied to \overline{IRQ}

15.3.1 Functional Description

Figure 15-9 shows a simplified diagram of monitor mode entry.

The monitor module receives and executes commands from a host computer. Figure 15-10, Figure 15-11, and Figure 15-12 show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

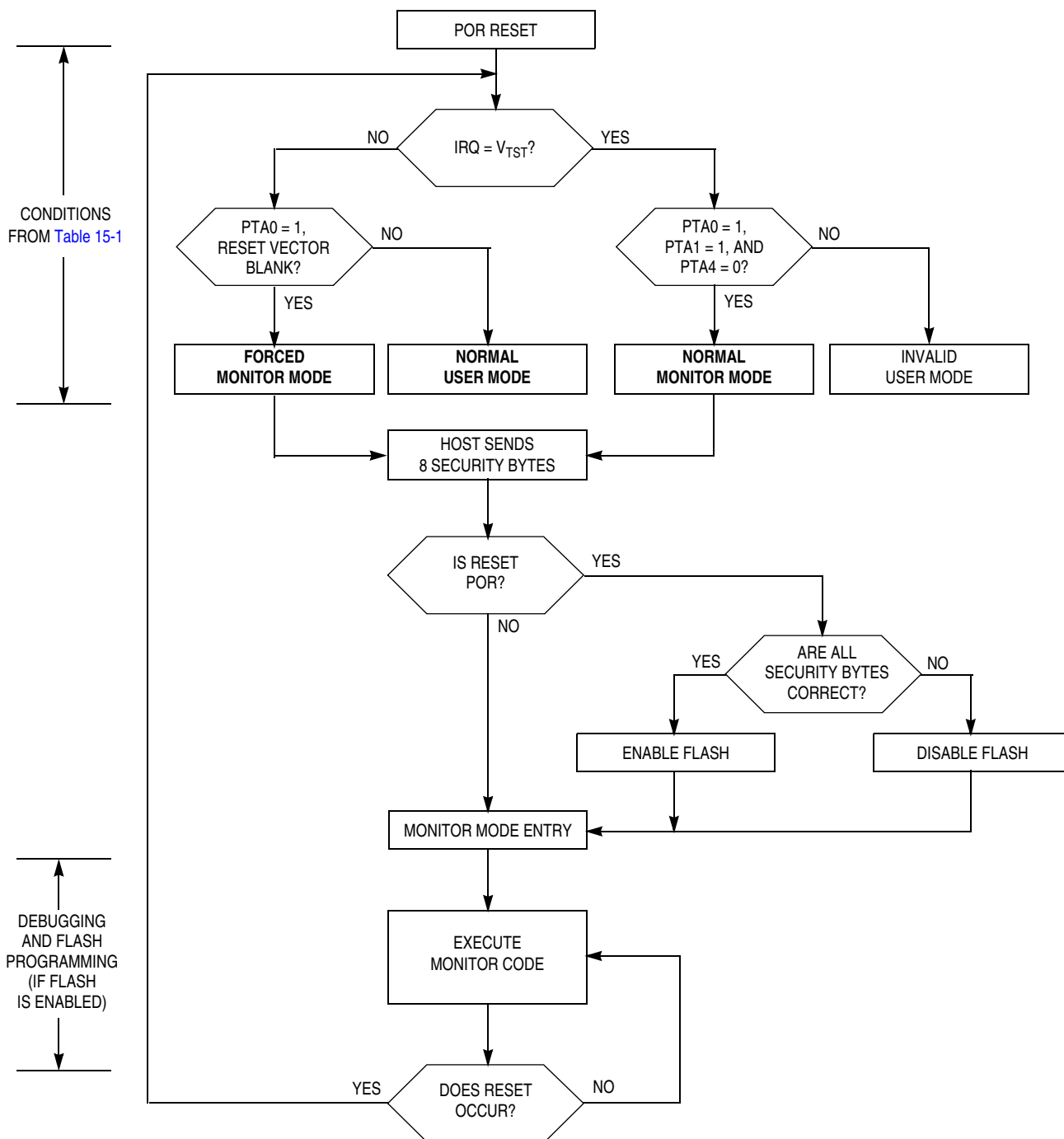


Figure 15-9. Simplified Monitor Mode Entry Flowchart

Chapter 16

Electrical Specifications

16.1 Introduction

This section contains electrical and timing specifications.

16.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to [16.5 DC Electrical Characteristics](#) for guaranteed operating conditions.

Characteristic ⁽¹⁾	Symbol	Value	Unit
Supply voltage	V_{DD}	-0.3 to +6.0	V
Input voltage	V_{IN}	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
Mode entry voltage, \overline{IRQ} pin	V_{TST}	$V_{SS} - 0.3$ to +9.1	V
Maximum current per pin excluding PTA0–PTA5, V_{DD} , and V_{SS}	I	±15	mA
Maximum current for pins PTA0–PTA5	$I_{PTA0} - I_{PTA5}$	±25	mA
Storage temperature	T_{STG}	-55 to +150	°C
Maximum current out of V_{SS}	I_{MVSS}	100	mA
Maximum current into V_{DD}	I_{MVDD}	100	mA

1. Voltages references to V_{SS} .

NOTE

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{IN} and V_{OUT} be constrained to the range $V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD} .)

SHEET: 1 OF 3